IMPACT: International Journal of Research in Engineering & Technology
ISSN(P): 2347–4599; ISSN(E): 2321–8843
Vol. 8 Issue 4, Apr 2020, 7–28

SCHEDULING OF TASK IN CLOUD USING A DIFFERENT ALGORITHM & WORKING OF MODIFIED PSO

Babita Bhagat[1] & P.Sayanasi Naidu[2]

[1]*Research Scholar, Pillai HOC College of Engineering and Technology, Rasayani, Khalapur, Maharashtra, India*

[2]*Research Scholar, Department of Computer Science & Engineering, GITAM University, Visakhapatnam, Andhra Pradesh, India*

Received: 02 Apr 2020 **Accepted:** 10 Apr 2020 **Published:** 20 Apr 2020

ABSTRACT

Right now seeming near the research of the particular calculations for their sensibility, probability, adaptability regarding the matter of cloud situation, after that we attempt to suggest the hybrid technique that might be grasped to improve the contemporary level further. With the goal that it can move cloud suppliers to offer a superior fine of administrations. A few metaheuristic calculations which incorporate Particle Swarm Streamlining (PSO) technique and Variable Neighborhood Particle Swarm Enhancement (VNPSO) system are utilized to address the business venture organizing issue in flowed enrolling. While circulating assignments to the machines, to fulfill the wellbeing conditions and to limitation as far as possible, we proposed a changed PSO with a scout adjustment (MPSO-SA) calculation which utilized a cyclic term called change government to get the charming worth breaking point. The presentation of the proposed MPSO-SA booking component is separated and the Genetic calculation (GA), PSO and VNPSO structures and the exploratory final product flaunt that the proposed technique diminishes the opportunity of risk with security conditions and it has perfect intermixing resources over the current shows.

KEYWORDS: *Cloud Computing, Met Heuristic, GA, PSO and VNPSO*

INTRODUCTION

Cloud computing has become one of the most important services which are being utilized by both technical and non-technical people. Non-technical users who are not even aware of the concepts of cloud computing are taking advantage of the cloud services by making their computational tasks handled by the cloud. In order to gain the least execution time, the development of the Cloud applications is executed in parallel. Cloud can be either public or private. In the public clouds, the data centres are managed and monitored by a third party who is why security turns out to be a major issue. In private cloud, security is not compromised since the data centre is within an organization.

As the number of cloud users is increasing with time, the need for providing a quality service has become of great importance. In this aspect, the notion of scheduling becomes an important task which is to be provided to the user. Scheduling refers to the execution sequence of the tasks which are allocated to machines. By the means of scheduling, one of the most important qualities of service, i.e. latency is minimized. [44] The cloud cannot restrict its working like the conventional methods (Distributed Systems) so that is why it is dynamic in nature. This implies that the scheduling algorithms should also be dynamic and not static. The dynamic scheduling differs from static in a way that the prior

information regarding the incoming tasks and the number of resources available for them, is known. Scheduling can be implemented using two methods- one is by scheduling the VM and the other is by scheduling the host.[44] The latter is implemented by scheduling the number of VM's considering the requirements of the host. The scheduling by VM will depend on the incoming tasks and the VM assigned to the user serves as an input container for their data.

In this paper, the concept of task scheduling and its classification are discussed along with the comparative study of various task scheduling algorithms on the basis of various qualities of service parameters. Later, the advantages and disadvantages of these scheduling algorithms are also mentioned

Related Work

Advancement issues are in Class NP-hard. These issues can be understood by the identification technique, heuristic strategy or guess strategy. In the identification strategy, an ideal arrangement can be chosen if all the potential arrangements are counted and looked at individually. At the point when various examples are huge, the comprehensive count isn't possible for planning issues. All things considered, a heuristic is a problematic calculation to discover sensibly great arrangements sensibly quick. Guess calculations are utilized to discover surmised answers for an enhanced arrangement. These calculations are utilized for issues when careful polynomial-time calculations are known. Improving assignment information territory is huge scope information preparing frameworks is pivotal for work culmination time. The greater part of the ways to deal with improve information territory are either avaricious and overlook worldwide enhancement, or experience the ill effects of high calculation unpredictability. This issue is tended to by proposing a heuristic errand booking calculation called Balance-Reduce (BAR) in [1].

Burden adjusting task scheduler balances the whole framework load while attempting to limit the make span of given undertakings set. Two diverse burden adjusting planning calculations dependent on subterranean insect settlement are proposed in [2] and [3]. Another subterranean insect state-based calculation intends to limit work culmination time dependent on pheromone is proposed in [9]. Cloud Loading Balance calculation [10], adds the ability to the dynamic equalization component for the cloud condition. The choice, which remaining tasks at hand to redistribute to what cloud supplier, ought to expand the use of the inside framework and limit the expense of running the re-appropriated undertakings in the cloud while considering the applications' nature of administration limitations. A lot of heuristics, to cost-proficiently plan cut off time obliged computational applications, is proposed in [11]. The multi-objective meta-heuristics planning calculation for a multi-cloud condition is proposed in [12]. This calculation attempts to accomplish application high accessibility and adaptation to non-critical failure while lessening the application cost and keeping the asset load expanded. In light of the expanding enormous Web chart and interpersonal organizations, cost-cognizant huge diagram handling planning is significant and a heuristic for the equivalent is proposed in [13]. An advanced calculation dependent on GA to plan autonomous and detachable errands adjusting to various calculations and memory necessities is proposed in [14]. Multi-operator hereditary calculation (MAGA) [15] is a half and half calculation of GA which takes care of the heap adjusting issue in distributed computing. COA (Course Of Activity) arranging includes asset designation and assignment planning. A strong COA arranging with fluctuating spans dependent on GA is proposed in [16].

Lessening vitality utilization is an undeniably significant issue in distributed computing, all the more explicitly when managing High-Performance Computing (HPC). A multi-objective hereditary calculation (MO-GA), proposed in [17], upgrades the vitality utilization, carbon dioxide outflows and the created benefit of a geologically appropriated distributed computing

framework. Another equal hereditary calculation based asset planning is proposed in [18]. Re-enacted strengthening is a conventional probabilistic metaheuristic for the worldwide improvement issue of finding a decent estimate to the worldwide ideal of a given capacity in an enormous pursuit space. An improved calculation for task booking dependent on a hereditary re-enacted strengthening calculation in distributed computing is proposed in [19]. The versatility of a figuring framework can be mostly distinguished by size, geological circulation, managerial limitations, heterogeneity, vitality utilization, and straightforwardness. A low intricacy vitality proficient heuristic calculation for planning, proposed in [20], performs productively exhibiting their materialness and adaptability. In clump mode, errands are planned uniquely at some predefined time. This empowers bunch heuristics to think about the genuine execution times of a bigger number of errands. Min-min and Max-min are heuristics utilized for bunch mode planning. Heuristics based improved Max min calculation is proposed in [21] and the QoS Min-Min booking calculation is proposed in [22]. Pack of undertakings (BoT) applications is the one that executes autonomous equal errands. Heuristics proposed in [23] intends to augment asset use while executing BoTs in heterogeneous arrangements of Cloud assets distributed for various quantities of hours. Another spending limitation scheduler proposed in [24] plans huge packs of errands onto various mists with various CPU execution and cost, limiting culmination time while regarding an upper destined for the financial backing to be spent. At the point when suppliers can't reveal private data, for example, their heap and processing power, which are generally heterogeneous, the Meta scheduler needs to settle on dazzle booking choices. Right now, the cutoff time obliged BoT application planning approach is proposed in [25].

Hai Zhong1, Kun Tao1, Xuejie Zhang [26] proposed a streamlined planning calculation to accomplish the enhancement or sub-improvement for cloud booking. Right now, the Improved Genetic Algorithm (IGA) is utilized for the robotized planning approach. It is utilized to build the usage pace of assets and speed. Suraj Pandey, LinlinWu, Siddeswara Mayura Guru, Rajkumar Buyya [27] introduced a molecule swarm advancement (PSO) based heuristic to plan applications to cloud assets that consider both calculation cost and information transmission cost. It is utilized for work process application by fluctuating its calculation and correspondence costs. The trial results show that PSO can accomplish cost reserve funds and a decent circulation of the outstanding tasks at hand onto assets. Reference [28] explored the viability of rescheduling utilizing cloud assets to expand the unwavering quality of employment fulfilment. In particular, plans are at first created utilizing framework assets while cloud assets are utilized distinctly for rescheduling to manage delays in work fruition. A vocation in their investigation alludes to a pack of-assignments application that comprises of countless free errands; this activity model is regular in numerous sciences and building applications. They have formulated a novel rescheduling strategy, called rescheduling utilizing mists for solid finishing and applied it to three notable existing heuristics.

Indeed, the task has been seen as NP-complete [29]. Since the task is an NP-Complete issue, the Genetic Algorithm (GA) has been utilized for the task [30]. However, the hereditary calculation may not be the best strategy. Reference [31] has represented that the molecule swarm enhancement calculation can show signs of improvement plan than the hereditary calculation in-network processing. Reference [32] has demonstrated that the presentation of the Particle Swarm Optimization (PSO) calculation is better than the GA calculation in a circulated framework. Not exclusively is the PSO calculation arrangement quality better than GA in the majority of the experiments, yet in addition, the PSO calculation runs quicker than GA. Along these lines, we utilize a strategy called Particle Swarm Optimization to streamline the assignment booking issue. Right now, centre around limiting the all-out executing time and moving time. Meng Xu, Lizhen Cui, Haiyang Wang, Yanbing Bi [33] chipped away at various work processes and numerous QoS. They executed a methodology for numerous work process the executive's frameworks with different Quality of Service. The entrance rate

for booking is expanded by utilizing this methodology. This technique limits the make span and cost of work processes. Topcuogluet. al, [34] introduced the HEFT calculation. This calculation finds the normal execution time of each undertaking and furthermore the normal correspondence time between the assets of two errands. At that point undertakings in the work, the process is requested on a position work. At that point, the assignment with higher position esteem is given a higher need. In the asset determination stage, errands are planned for needs and each undertaking is relegated to the asset that finishes the assignment at the most punctual time.

Salehi, M.A., and Buyya, R. [35] proposed a market-situated progressive planning methodology which comprises of both assistance levels booking and undertaking level planning. The administration level booking manages the Task to Service task and the errand level planning manages the enhancement of the Task to Virtual Machine task in neighbourhood cloud server farms. Yu, J. Buyya, R. what's more, Them, C.K. [36] proposed a cost-based work process booking calculation that limits the execution cost while complying with the time constraint for conveying results. It can likewise adjust to the deferrals of administration executions by rescheduling unexecuted errands. Sakellariou, R., Zhao, H., Tsiakkouri, E. what's more, Dikaiakos, M.D [37] proposed a fundamental model for work process applications that displayed as a coordinated non-cyclic chart (DAGs) and that permit to plan the hubs of DAG onto assets in a manner that fulfils spending imperative and is upgraded for by and large time.

Burke et al [38] propose a hyper-heuristic structure that executes ordinarily utilized chart shading heuristics combined with an arbitrary requesting heuristic. Tabu inquiry is utilized as the elevated level quest strategy for creating great successions of low-level heuristics. Every heuristic rundown delivered by the tabu pursuit calculation is assessed by successively utilizing the individual heuristics to arrange the unscheduled occasions and hence build a total timetable. This work likewise features the presence of two hunt spaces: the heuristic space and the issue arrangement space. The methodology was tried on both course and test timetabling benchmark occurrences with serious outcomes. A subsequent paper [39] looks at the exhibition of a few metaheuristics that work on the hunt space of heuristics. Iterative procedures, for example, iterated nearby inquiry and variable neighbourhood search were seen as increasingly compelling for navigating the heuristic hunt space. The examination likewise actualized hybridizations of the hyper-heuristic structure with standard nearby hunt working on the arrangement space, which was found to improve the exhibition of the general framework, making it serious with best in class approaches on the considered benchmark occasions. A further report [40] utilizes the thought of wellness scenes to break down the inquiry space of chart shading heuristics. These scenes are found to have an elevated level of lack of bias (i.e., the nearness of levels). Besides, albeit tough, they have the empowering highlight of an all-around curved or huge valley structure, which shows that an ideal arrangement would not be separated yet encompassed by numerous neighbourhood minima. Li et al [41] examine two information mining procedures, fake neural systems, and parallel calculated relapse to discover worldwide examples covered up in huge informational collections of heuristic successions. With the prepared characterization governs, the exhibition of a subsequent arrangement during the hyper-heuristic inquiry can be anticipated without the need to embrace the computationally costly assurance of the arrangement and estimation of the goal work. In the underlying examination [42], every component of the populace is a variable-length string, where each character speaks to a heuristic. The methodology created practical assessment timetables with delicate limitations inside the scope of other quest techniques utilized for this reason and beat past hyper-heuristics on some of the tried occasions. The thought is to learn the relationship between issue states and sufficient heuristics for timetabling. In particular, the framework attempts to find a lot of name focuses on the space of the issue states.

Each name alludes to a heuristic, and the calculation works by over and again finding the closest named point to the present condition and applies its mark until a total arrangement is constructed. Different various types of issue state portrayal and strategies for estimating the wellness were contemplated. The methodology had the option to create quick and basic critical thinking calculations that offer great exhibition over a scope of tests and class timetabling issues. Sabar et al [43] use progressive hybridizations of four low-level chart shading heuristics for delivering even orderings. A consolidated trouble list is determined by considering all the orderings and occasions are planned by this record. The methodology created a serious outcome in the examined benchmark occurrences. Quantities of creators have accomplished work in the zone of planning calculations. Table 1 speaks to the relative investigation of a different planning calculation, nature of booking calculation, target criteria, for example, the parameters which have been engaged for improvement and the earth in which the planning calculations were applied. The heuristic calculations are need-based and for the most part, issue-driven. The engineer can utilize his own understanding to relegate the need to work process applications and cloud assets.

In 2015, Lingfang Zeng kept an eye out for the issue of security in work process booking and present a Security-Aware and Budget-Aware work process orchestrating strategy (SABA) to get shorter make span and checked to organize under some spending essential. They besides orchestrated a fit framework to pack errands subject to information reliance that decreases the expense and time of recovering information in the cloud structure. The primer results showed that SABA is sensible and productive in making cost and execution of work process booking; additionally, gives better security in cloud condition.

In 2016, Zhongjin proposed security and cost cautious booking (SCAS) figuring for steady work structures that consider the heterogeneous undertaking with various qualities like information concentrated, memory-authentic or calculation raised. SCAS figuring is, for the most part, used to limit the expense of execution at which it faces the due date and threat rate requirements. They utilized the coding approach of PSO to discover the answer for a multi-essential and multi-dimensional streamlining issue in the work process orchestrating.

In 2016, Saima Gulzar Ahmad has changed an intrinsic assessment based strategy to another crossbreed acquired figuring (HGA) that gives an ideal blueprint to less extent of the time. This calculation uses the advantages in a tremendous aggregate by updating the store change at the time of execution. At long last, the presentation of HGA calculation is investigated utilizing sorted out datasets and genuine application work structures.

In 2015, Mala Kalra and Sarbjeet Singh have introduced an outlined methodology for booking in conveyed processing. A minimum necessity assessment and near assessment of different organizing tallies, generally, the three unavoidable metaheuristic estimations: Ant Colony Optimization (ACO), GA and PSO, and two novel systems: League Championship Algorithm (LCA) and BAT figuring are examined at the present time.

In 2016, Mohammed Abdullahi has discovered an ideal reaction for able undertaking organizing utilizing Symbiotic Organism Search (SOS) which is a metaheuristic improvement philosophy. The SOS mean endeavour masterminding is executed utilizing makes pan, reaction, and level of lopsidedness among different virtual machines. At long last, the quantifiable reports of the outcomes are separated and SAPSO utilizing some tremendousness test.

Classification of Scheduling

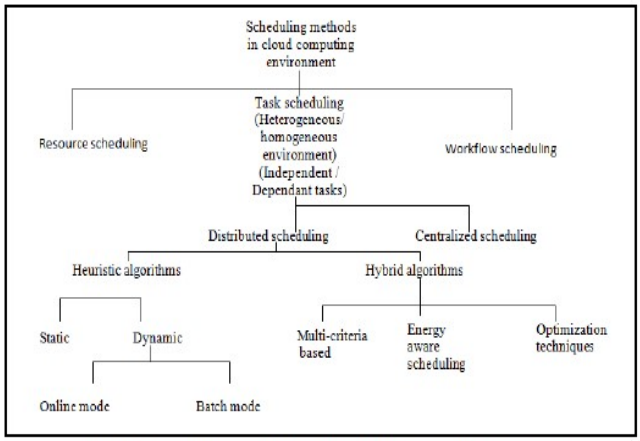

Fig. 1. Classification of Scheduling Methods in Cloud Computing Environment

Figure 1

Classification of Scheduling

By and large, planning for distributed computing is arranged into various classes. The principal class depends on Task. In light of the assignment, booking is separated into static planning and dynamic booking. In static planning task shows up at the same time at the processor and the errands are submitted¬ on the accessible assets, booking choices are taken before assignments are submitted. The handling time is refreshed after errand finish; this sort of assignment based booking is, for the most part, applied for the intermittent undertaking. On account of the dynamic booking number of the undertaking, the machine area, and asset distribution are not fixed. Appearance times of the undertakings are not known before accommodation. Further, powerful booking is grouped into two kinds either by cluster mode or online mode. In bunch, mode assignments are lined, gathered into a set and booked after a fixed timeframe. In online mode, task is planned when they show up in the framework.

The subsequent classification depends on different measurements utilized in the cloud and characterized into a group framework, intuitive framework, and continuous framework. In the bunch framework turnaround, time and throughput can be determined. The reaction time and reasonableness can be determined to utilize an intuitive framework and in a constant framework, the cut-off time is estimated. The third class is execution and market-based. In the exhibition put together concentration with respect to ideal execution time not thinking about cost, certain arrangements are considered for mapping the errand and execution. In the market-based scenario, simply think about the expense as the factor. The backtracking calculation, Genetic Algorithm depends on this market-based booking calculation. In static planning, all customary booking calculations can be executed FCFS, cooperative effort, min-min, and max-min. In unique planning, all heuristic booking calculations can be executed in the Genetic Algorithm, Particle Swarm Optimization, Simulated Annealing, Ant Colony Optimization, and dynamic rundown planning.

Dynamic errands are for the most part spoken to by coordinated non-cyclic diagram; different measurements used to ascertain the quality of administration in the cloud are make span, early finish time, complete execution time, cost, trust, reasonableness, cut-off time, reaction time, unwavering quality and accessibility.

Overview of Task Scheduling

Cloud comprises of various assets that are particular with one other by methods for a couple of means and cost of performing undertakings in the cloud utilizing resources of the cloud is differing, so booking of assignments in the cloud is one of a kind according to the standard techniques of planning thus booking of errands in the cloud need better insightfulness in regards to being paid in light of the fact that administrations of the cloud depend upon them. Undertaking booking has a key impact on improving the adaptability and steadfast nature of frameworks in the cloud. The rule clarification for planning errands to the benefits according to the given time-bound, which incorporates finding an aggregate and best course of action in which various endeavours can be executed to give the best and pleasing result to the customer. In distributed computing, assets in any casing for example holders, firewalls, arrange are reliably continuously dispensed by the progression and necessities of the assignment, subtasks. Thus, this prompts task planning for the cloud to be a unique issue suggests no earlier described progression may be useful during the handling of assignments [20]. The purpose for the planning to be dynamic is that since the surge of an errand is uncertain, execution ways are also questionable and simultaneously assets open are similarly mysterious in light of the fact that there are different assignments that are accessible that are sharing them simultaneously meanwhile. The booking of endeavours in the cloud intends to pick the best fitting resource available for the execution of assignments or to designate PC machines to tasks in such a manner, to the point that the completing time is constrained as would be reasonable. In planning calculations, once-over of endeavours is made by offering a need to each and every task where the setting of need to various errands can be established on various parameters. Undertakings are by then picks according to their requirements and given out to accessible processors and PC machines that satisfy a predefined target work [12].

Booking Types

- Static planning plan undertakings in the known condition, for example, it as of now has the data about complete structure of undertakings and mapping of assets before execution, appraisals of assignment execution / running time.
- Dynamic booking must rely upon not just the submitted assignments to a cloud situation yet additionally the present conditions of a framework and PC machines to settle on a planning choice. Distributed computing utilizes a virtualization system for mapping the assets of cloud to the virtual machine layer, actualize the client's errand, so the assignment booking of distributed computing condition accomplishes at the applications layer and the virtual layer of assets [19]. Booking is only the mapping of errands and assets as per some specific standards for accomplishing the ideal objective. Distributed computing worldview streamlines the mapping of errands to assets; the necessary assets together structure to be Virtual Machines (VMs), the procedure of search the ideal asset bundle is equivalent to the way toward looking the different VMs.

Various Scheduling Algorithm

Following Booking Calculations are Right Now Common in Mists

- Resource-Aware-Scheduling calculation (RASA): Saeed Parsa and Reza Entezari-Maleki [2] proposed another undertaking booking calculation RASA. It is made out of two conventional planning calculations; Max-min and Min-min. RASA utilizes the benefits of Max-min and Min-min calculations and spreads its hindrances. In spite of the fact that the cut-off time for each undertaking, showing up pace of the errands,

cost of the assignment execution on every one of the assets, the expense of the correspondence are not thought of. The exploratory outcomes show that RASA is outflanked the current booking calculations in enormous scope circulated frameworks.

- RSDC (Reliable Scheduling Distributed In Cloud Computing): Arash Ghorbannia Delavar, Mahdi Javanmard, Mehrdad Barzegar Shabestari and Marjan Khosravi Talebi[13] proposed a dependable booking calculation in a distributed computing condition. Right now, work is separated into sub-employments. In order to adjust the employments, the ask for and recognize time are determined independently. The booking of each activity is finished by ascertaining the solicitation and recognizes time as a common activity. With the goal that the effectiveness of the framework is expanded.

- An Optimal Model for Priority-based Service Scheduling Policy for Cloud Computing Environment: Dr. M. Dakshayini, Dr. H. S. Guruprasad [14] proposed another planning calculation dependent on need and confirmation control plot. Right now, is doled out to each conceded line. Affirmation of each line is chosen by computing bearable deferral and administration cost. The upside of this calculation is that this arrangement with the proposed cloud design has accomplished a high (99 %) administration finishing rate with ensured QoS. As this approach gives the most noteworthy priority to generously compensated client administration demands, the general overhauling cost for the cloud additionally increments.

- A Priority based Job Scheduling Algorithm in Cloud Computing: Shamsollah Ghanbari, Mohamed Othman proposed another booking calculation dependent on multi-criteria and multi-choice need driving planning calculation. This booking calculation comprises of three degrees of planning: object level, characteristic level, a substitute level. Right now can be set by the activity asset proportion. At that point, a need vector can be contrasted and each line. This calculation has a higher throughput and less completion time.

- Broadened Max-Min Scheduling Using Petri Net and Load Balancing: El-Sayed T. El-kenawy, Ali Ibraheem El-Desoky, and Mohamed F. Al-rahamawy[15] have proposed another calculation dependent on the effect of RASA calculation. The improved Max-min calculation depends on the normal execution time rather than complete-time as a choice premise. Petri nets are utilized to display the simultaneous conduct of disseminated frameworks. Max-min shows accomplishing plans with practically identical lower make span as opposed to RASA and unique Max-min.

- An Optimistic Differentiated Job Scheduling System for Cloud Computing: Shalmali Ambike, Dipti Bhansali, Jaee Kshirsagar, Juhi Bansiwal [16] has proposed a separated planning calculation with non-pre-emptive need lining model for exercises performed by cloud client in the distributed computing condition. Right now web application is made to do some action like one of the documents transferring and downloading then there is a requirement for proficient occupation planning calculation. The QoS necessities of the distributed computing client and the most extreme benefits of the distributed computing specialist organization are accomplished with this calculation. Improved Cost-Based Algorithm for Task Scheduling: Mrs. S.Selvarani, Dr. G. Sudha Sadhasivam [17] proposed an improved cost-based planning calculation for making the proficient mapping of errands to accessible assets in the cloud. The spontaneous creation of

conventional action-based costing is proposed by another assignment planning procedure for a cloud domain where there might be no connection between the overhead application base and the way that various undertakings cause the overhead expense of assets in the cloud. This booking calculation partitions all client assignments relying upon the need for each errand into three unique records. This booking calculation estimates both asset cost and calculation execution, it additionally improves the calculation/correspondence proportion.

- Execution and Cost assessment of Gang Scheduling in a Cloud Computing System with Job Migrations and Starvation Handling: T. Mathew, K. Sekaran [18] has proposed a group booking calculation with work movement and starvation dealing within which planning equal employments, effectively applied in the zones of Grid and Cluster registering. The number of Virtual Machines (VMs) accessible at any minute is dynamic and scales as indicated by the requests of the employments being adjusted. The previously mentioned model is concentrated through recreation so as to break down the presentation and generally cost of Gang Scheduling with relocations and starvation dealing with. Results feature that this booking technique can be successfully sent on Clouds, and that cloud stages can be suitable for HPC or superior venture applications.

Table 1: Scheduling Algorithms

Scheduling Algorithm	Scheduling Method	Scheduling Parameter	Scheduling Factor	Findings	Environment
Resource-aware-scheduling	Batch mode	Makes pan	Grouped task	1-It is used to reduce the makes pan	Grid environment
RSDC (Reliable Scheduling Distributed In Cloud Computing)	Batch mode	Processing time	Grouped task	1-It is used to reduce processing time 2- It is efficient for load balancing	Cloud environment
An optimal model for priority-based service scheduling policy for the cloud computing environment	Batch mode	Quality of service request time	An array of workflow instances	1-High QoS 2-High throughput	Cloud environment
A priority based job scheduling	Dependency mode	Priority to each queue	An array of a job queue	1-Less finish time	Cloud environment
Extended max-min scheduling using Petri net and load balancing	Batch mode	Priority to each queue	An array of a job queue	1-It is used for efficient load balancing. 2-Petrin net is used to remove the limitation of the max-min algorithm	Cloud environment
An optimistic differentiated job scheduling system for cloud computing	Dependency mode	Quality of service max-min profit	A single job with multiple users	The QoS requirements of the cloud computing user and the max-min profits of cloud computing service provider are achieved	Cloud environment
Improved cost-based algorithm for task scheduling	Batch mode	Cost, performance	Unscheduled task group	1-measures both resource cost and computation performance 2-Improves the computation ratio	Cloud environment
Performance and cost evaluation of gang scheduling	Batch mode	Performance cost	Workflow with a large number of job	1-the application of migrations and starvation handling had a significant effect on the model 2-it improves performance	Cloud environment

Work Flow Application Model for Cloud System

Work Flow Architecture

Cloud condition includes an aggregation of virtual machines that give organizations, for instance, Infrastructure organization, Platform organization, and Software organization. Furthermore, it has some critical features like virtualization, apportionment, and adaptability. When in doubt, dispersed processing maps the applications to the cloud virtual machines using virtualization method thus reserving of endeavours for each mapped machine is finished in the application layer and the virtual layer. The arranging implies a lot of standards to orchestrate the endeavours to be performed by the composed frameworks. Here, the complexities of tremendous scope issues are diminished by work process application and subsequently work process booking finds prime importance. The essential purpose of the work procedure booking is to constrain the make span by dispensing certain assignments to explicit machines in a suitable way. Work process designing which appears in Fig. 1 includes a heap of parts that must be an arrangement for a back to the back solicitation.

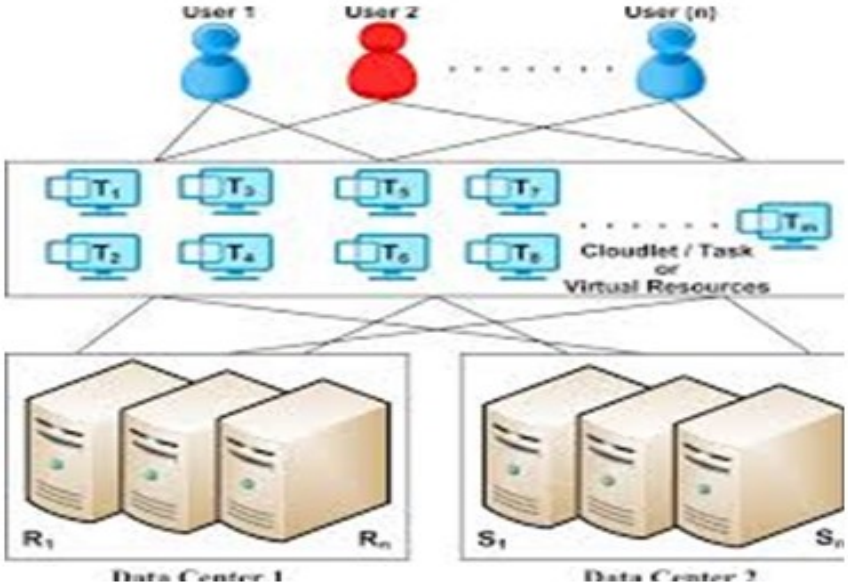

Figure 2: Workflow Architecture of Cloud Computing Environment.

System Model

The PC application in which systems a colossal volume of data is named as data heightened that gives a huge bit of its dealing with time to I / O and from that point forward, it controls the data. In data raised application, the action, and the advantages must meet the necessities of the technique stream, data movement and data find a good pace, satisfaction, security restrictions, versatility, and availability. The data heightened application is improved by the work procedure, which breakdown the application into a more diminutive task and is taken care of in a course of action solicitation to achieve the result.

In the data concentrated figuring condition, the data resource (database is related with the server) is related with the computational unit (Super-PC or some different machines) by an interfacing join which addresses the relating transmission limit and the constraint of the enrolling unit depends on the amount of central taking care of unit (CPU),

memory and additional room and other specialization. The planning speed of each machine is conveyed as the number of cycles per unit time.

Allow us to consider the data-packed application in cloud condition with five handling units and five data resources and all of the fragments must interface clearly or by suggestion with each other as showed up in Fig. 2. Be that as it may, the sections of data raised figuring can be summarized as preparing units in which each enlisting unit outfits the enrolling organization with the security rank and data resources in which each datum resource gives the data organization the security rank and both the security positions are usually named as.

A set of task/operation constitutes a job that has to be performed on a machine with crucial processing constraints. Three of such important processing constraints are as follows,

Work flow constraints: Arranging a set of task or operation for a specific application is referred as workflow and each task can be performed after the completion of the previous task in sequence.

Processing length: The number of cycles that require for the completion of an operation is termed as processing length. A task has to be performing within this processing length.

Security constraints: If a workflow application consists of K jobs $\{j_1, j_2, .., j_K\}$ and the q^{th} job consists of a set of task/operation $\{T_{q,1}, T_{q,2}, ..., T_{q,p}\}$. To reduce the difficulty, all the operations are combined together as $\{T_1, T_2, ..., T_{N_T}\}$, where N_T refers to the total number of tasks and it is determined using eq. (1),

$$N_T = \sum_{q=1}^{K} |T_q| \tag{1}$$

In the eq. (1), the cardinality term $|Tq|$ refers to a set of tasks in q^{th} job.

The security demand of the computing service SD_{cs,T_i} and the security demand of the data service SD_{ds,T_i} are commonly termed as S_T .

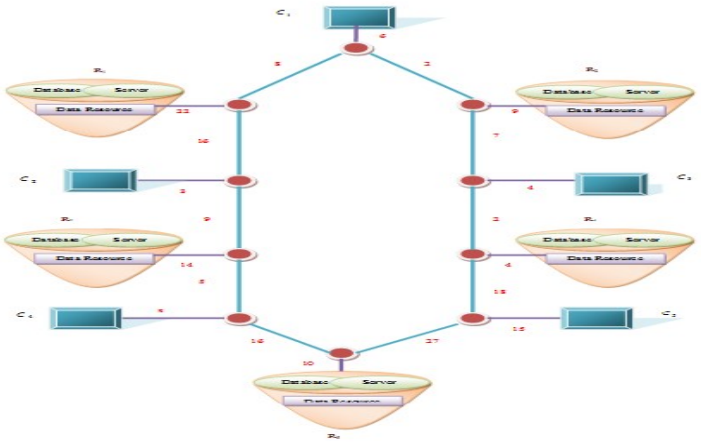

Figure 3:Data-Intensive Computing Environment With Five Computing Units And Five Data Resources.

Security Constraints

In the information concentrated figuring condition, three security modes are accessible for work booking process:

Secure mode: Generally, the activities that fulfill the security necessities and that are available on the figuring units are planned for this protected mode. This mode is a customary methodology for booking an occupation. Right now, the security request of the errand is lesser than or equivalent to the security rank of figuring unit or information asset, for example, at that point just the activity is apportioned to the comparing figuring unit or information asset.

Risky mode: All the achievable dangers are evacuated by planning the errand just to the accessible registering unit or accessible information asset. This mode is a serious and intense methodology for planning the activity.

Gamma Risky mode: In this mode, the task is booked to accessible information assets or machines taking all things considered a hazard, which speaks to the estimation of likelihood with and for the comparing secure and hazardous modes.

For the most part, the safe mode is over the top expensive to achieve, so hazard mode and - dangerous mode is usually utilized to tackle work booking issue. In the distributed computing condition, the security levels are drawn nearer by a subjective or fluffy scale which comprises of five levels in particular, exceptionally high, high, medium, low and very low.

The probability of risk for the security constraint model is defined in eq. (2) and it is graphically represented in Figure 4.

$$\Pr ob_{risk} = \begin{Bmatrix} 0 & if & S_T - S_C \le 0 \\ 1-e^{-0.5(S_T-S_C)} & if & 0 < S_T - S_C \le 1 \\ 1-e^{-1.5(S_T-S_C)} & if & 1 < S_T - S_C \le 2 \\ 1 & if & 2 < S_T - S_C \le 5 \end{Bmatrix} \quad (2)$$

A figuring unit with the condition, or is considered as a protected machine and here, the likelihood of hazard is zero. On the off chance that a planned activity is relegated to a machine with the condition, at that point the likelihood of hazard must be lesser than half and in the event that an assignment is given to a registering unit with, at that point the booked errand will be done by the machine. The booked activity is executed later, if, yet it will be executed before as far as possible. At the point when the assignment can't be practiced and it must be rescheduled.

In this condition, planning has turned into a significant worry to perform in the information escalated cloud condition.

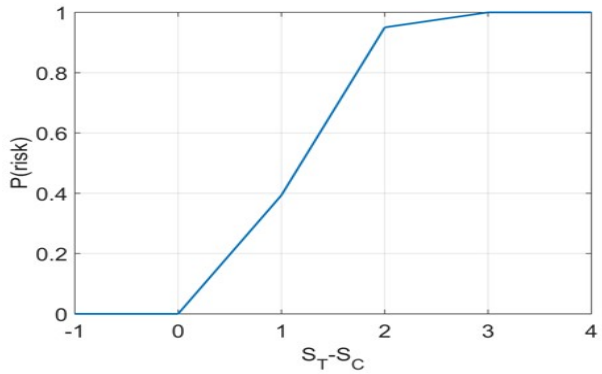

Figure 4: Graphical Representation of Risk Probability.

SECURE WORK FLOW SCHEDULING

Scheduling Model

Let us consider a work flow application that is executed by the machine with the processing speed $\{Ps_1, Ps_2,.., Ps_{N_c}\}$ and the processing length $\{l_1, l_2,..., l_{N_T}\}$ and finally it is subjected to a set of security constraint, $CS = \{S_T, S_C\}$.

Figure 5 shows the Directed Acyclic Graph (DAG) based workflow representation for job scheduling. The workflows are usually designed as Directed Acyclic Graphs (DAGs), $G\{T, E\}$ in which vertices, $T = \{T_1, T_2,..., T_{N_T}\}$ denotes the individual task of the workflow and edges E of the graph denotes the precedence or data dependencies among the task.

Definition 1: An edge of the DAG is addressed as, which surmises that the task can't start to process until the endeavour completes its methodology and send the result to. Permit us to consider, is a great deal of precursor of the action and is a ton of successor of the task. Some supposition that is made for work making arrangements for handling condition, as it is an NP-troublesome issue,

- A successor task starts to perform following its forerunner completes its methodology, as it shows the availability of the machine.
- Only one endeavour/assignment can be dealt with by a machine at some random minute.
- Each task must be dealt with by any of the machines.
- The endeavour can't be reprocessed, when it completes its technique viably.
- The range for the course of action and move are uniform.
- The exercises are free.

Figure 5 Shows, the relations between the assignments are associated with a stream structure. If the edge exists in the graph, by then assemble its weight, if it doesn't exist, by then is set to.

The data resource states of assignments are found by the recuperation cross-section, in which the part shows the recuperation time for the task which is setting up the data recuperation from the data resource. Moreover, the throughput rates are controlled by the two structures and, in which the part addresses the restriction of the affiliation interface between the handling units or machines and, and the constraint of the affiliation associate between the enlisting unit and data resource is addressed by. The strategy fulfilment time for each undertaking is evaluated by the aggregate of the range for the three activities: time for recuperating data, the time is taken to incorporate data and the normal time to execute the movement on the open machine.

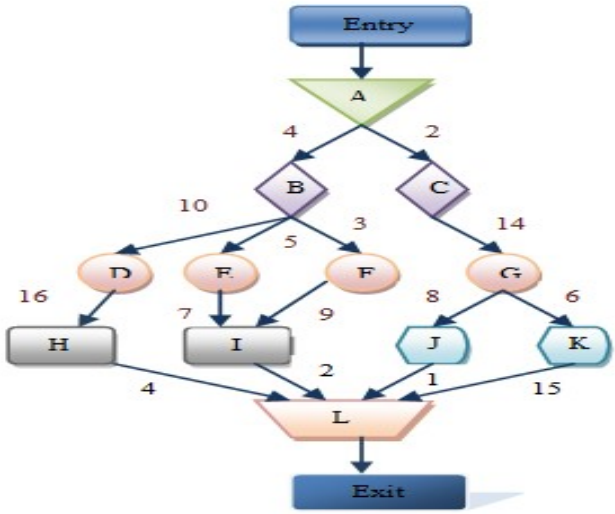

Figure 5: DAG Based Workflow Representation For Job Scheduling.

Let us consider a feasible solution, $Y = \{Y_1, Y_2, .., Y_{N_T}\}$, in which Y_i indicate the serial number of the computing unit to which the task T_i is allocate, then the completion time CT_{T_i} of the task T_i on the machine C_{Y_i} is evaluated using the equation (3)

$$CT_{T_i} = \sum_{\substack{x=1 \\ f_{x,i} \neq -1}}^{N_T} f_{x,i} u_{Y_x, Y_i} + \sum_{z=1}^{N_d} r_{i,z} v_{Y_i,z} + l_i / Ps_{Y_i} \quad (3)$$

The maximum completion time (makes pan) $CT_{\max}$ is computed using the equation (4),

$$CT_{\max} = \max\{\sum CT_{C_i}\} \quad (4)$$

The flow time i.e, the sum of the completion time of the solution is evaluated using the equation (5),

$$CT_{sum} = \sum_{i=1}^{c} \left(\sum CT_{C_i}\right) \quad (5)$$

The above equation (4) & equation (5) are used as a performance criterion for job scheduling problem in computing environment. If CT_{sum} gets minimize, it tends to reduce the execution time, and so the largest task requires a long time to process, whereas minimizing $CT_{\max}$ seeks to increase the execution time, and hence no task requires too long time to execute. As a result, to obtain an equal balance between these two criteria, a weighted aggregation is employed as in equation (6),

$$f = W_1\{CT_{\max}\} + W_2\{CT_{sum}\} \quad (6)$$

In the equation (6), W_1 and W_2 are non-negative weights, this can be fixed or adjusted but the sum of both weights should be unity.

Conventional Heuristics

- **Canonical PSO model:** The standard PSO count was animated from the direct of winged creature running and fish coaching. The two essential factors that depict the status of a particle in the interest space are its position and speed. Eq. (7) is used to revive the speed and Eq. (8) is used to choose the new situation by including the past position and the new speed. $h_{iq}(t+1) = Ih_{iq}(t) + ac_1 r_1\left(g_{iq}^{\neq}(t) - g_{iq}(t)\right) + ac_2 r_2\left(g_q^*(t)\right) - g_{iq}(t)\ )(7)$

$$g_{iq}(t+1) = g_{iq}(t) + h_{iq}(t+1) \tag{8}$$

In equation (7) and equation (8), and are the speed of the atom, the situation of the atom, past best position and the best region among all particles in the people individually. Here and are the discretionary limits inside a range and are the two positive steady parameters called speeding up coefficients, which decrease the best development size of the particle. The charming results can be procured by picking with and mean the inactivity factor while propelling the computation it's worth gets decreases from 1.0 to 0.

In order to screen the particle in chase space zone, the most extraordinary improvement of speed in one cycle and the circumstance of the atom is limited inside the range as in eq. (9) and eq. (10) as, $h_{iq} = sign\left(h_{iq}\right)\min\left(\left|h_{iq}\right|, h_{\max}\right)(9)$

$$g_{i,q} = sign\left(g_{i,q}\right)\min\left(\left|g_{i,q}\right|, g_{\max}\right) \tag{10}$$

Here, $h_{\max}$ is the maximum velocity and $g_{\max}$ is the maximum solution variable.

Generally, the PSO algorithm has two basic updating models namely *gbest* model and *pbest* model. According to eq. (7), the global best *(gbest)* position $g_q^*(t)$ is the best particle (obtained by means of fitness function) till the t^{th} iteration and the local best solution *(pbest)* $g_{iq}^{\neq}(t)$ is the best i^{th} particle in the t^{th} iteration and it is given in eq. (11),

$$g_i^{\neq} = \arg\min_{i=1}^{N}\left(f\left(g_i^{\neq}(t-1)\right), f\left(g_i(t)\right)\right) \tag{11}$$

The *gbest* and *pbest* along with r_1 and r_2 control the effect of new particle velocity. Generally, *gbest* model converges quickly than the *pbest* model, and also it has the tendency to confine in local optima but pbest model has the tendency to flow around the local optima.

- **Variable neighbourhood particle swarm optimization algorithm (VNPSO):** In equation (7) and equation (8), and are the speed of the atom, the situation of the particle, past best position and the best territory among all particles in the people separately. Here and are the self-assertive limits inside a range and are the two positive predictable parameters called speeding up coefficients, which diminish the best development size of the atom. The appealing results can be gained by picking with and mean the idleness factor while propelling the estimation its worth gets reduces from 1.0 to 0.

In order to screen the atom in chase space zone, the most extraordinary advancement of speed in one cycle and the circumstance of the particle is limited inside the range as in equation (9) and eq. (10)as, $ve_{iq}(t) = Wve_1 + ac_1 r_1\left(g_{iq}^{\neq}(t-1) - g_{iq}(t-1)\right) + ac_2 r_2\left(g_q^*(t-1) - g_{iq}(t-1)\right)(12)$

$$ve_1 = \begin{Bmatrix} ve_{iq} & if & |ve_{iq}| \geq ve_t \\ u_d ve_{\max} / \rho & if & |ve_{iq}| < ve_t \end{Bmatrix} \quad (13)$$

In equation (13), speaks to uniform arbitrary circulation work, is the most extreme speed and is the speed of molecule with a measurement. The estimation of two parameters and are related to the exhibition of the calculation. On the off chance that is enormous, at that point, the time of wavering gets diminished and it builds the likelihood that the particles bounce over the nearby minima for a similar number of emphases and furthermore it keeps the molecule from merging by keeping the particles in the quick flying state. These procedures are rehashed until it fulfils the end criteria.

Proposed MPSO-SA

The proposed MPSO-SA contrasts from PSO and VNPSO by demonstrating the scout change process after the new course of action centres are settled. Allow us to begin the self-assertive speed and sporadic situation for each atom with the estimation. By then, the wellbeing limit of the position can be settled for each particle. If the position well worth is lesser than the health limit of the local best for each particle then the position is close by the best position. Now and again, the wellbeing limit of the position ends up increasingly critical, then the local best health regard then another technique is proposed to register the number of positions to be changed using eq. (14), $N_{position} = \max\left\{\left[\frac{cycle}{N_{cycle}} * q\right], 1\right\}$ (14)

In the equation (14) is the adjust manager that gives the estimation of the closest entire number by leaving the sign and fragmentary part, where the amount of cycles is portrayed in set number. From the start, it is set to zero and expanded by one to gain the best position. Contingent on the number of positions, the change strategy occurs between the got position and the sporadic position and from that point onward, the new position gets refreshed. At that point, the method discussed in the endorsed PSO model has happened to get the best position, which differs in finding the overall best situation as given in equation (15)

$$gbest = \min\{f(pbest_i)\} = pbest_i^* \forall i \quad (15)$$

The Pseudo Code of the Proposed MPSO-SA Algorithm is illustrated in Algorithm 3 As Given Below:**Simulation Setup**

The propagation for the business arranging process in disseminated registering conditions and the execution of proposed MPSO-SA computation are done in MATLAB R2015a. The results are differentiated and various computations (GA, PSO, and VNPSO) to evaluate the sufficiency of the proposed MPSO-SA model with respect to get together, cost statics, and risk probability. The booking issues have been re-enacted in three cases by moving various jobs and figuring machines. Give the issue of booking 20 businesses in 5 figuring units an opportunity to be case 1 and arranging 50 occupations in 10 handling units be case 2 and booking 100 vocations in 20 enlisting units are case 3. The trial has been continued running in three cases for different occasions in excess of 100 cycles with different unpredictable commitments for each count. The multiplication is executed by fixing the characteristics as,

a) The size of the particle is twice the number of task/operation.

b) Inertia weight I is set to 0.9

c) The acceleration coefficients, $ac_1 = ac_2 = 1.49$

d) Set $cycle = 0$

Algorithm 1: Pseudo Code of Proposed Mpso-Sa Algorithm

Input : N, I, ac_1 and ac_2, r_1 and r_2, h_i, g_i and $cycle = 0$
Output: $gbest$
Begin
  Initialize N number of particles
  For each particle $i \in N$ and dimension D,
    Initialize velocity h_i and position g_i randomly
    Evaluate g_i by determining $f(g_i)$
    if $f(g_i) < f(pbest_i)$ for each particle i, then
      $pbest_i = g_i$ for each particle i
      reset $cycle = 0$
      else
        $cycle = cycle + 1$
        Find the number of position $N_{position}$ using equation (14) and update the new position g_i.
    end if
  If $\min\{f(pbest_i)\} \forall\ i < f(gbest)$ then
    $gbest = pbest_i^*$
  end if
  Update the particle's position and velocity using equation (7) and equation (8)
  end for
  $t = t + 1$
  Until $t > Max\_iterations$

Statistical Analysis

In this section, the statistical information of the existing and the proposed system is analyzed and the results are tabulated in Table I**,** Table II and Table III. The statistical analysis of all the algorithms in terms of cost function has reported that the mean value of MPSO-SA of case 1, case 2 and case 3 are 40.74%, 12.45% and 34.01% better than the conventional PSO model respectively.

Figure 6 shows analysis; it is clear that the best solution for the job scheduling problem is obtained by the proposed MPSO-SA algorithm than the existing algorithms.

Table 2

	Case 1		Case 2		Case 3	
	Proposed MPSO-SA Cost Analysis	**MPSO With Security Constraints**	**Proposed MPSO-SA Cost Analysis**	**MPSO With Security Constraints**	**Proposed MPSO-SA Cost Analysis**	**MPSO With Security Constraints**
Best	10.23	0.39347	939.16	25.93	420	12.997
Worst	115.68	3.5241	1103.1	29.55	557.68	21.373
Mean	78.631	2.3344	1018.8	27.411	468.59	15.8
Median	90.576	2.7869	1013.2	26.799	462.2	14.98
Standard deviation	41.429	1.3311	58.319	1.496	53.103	3.2469

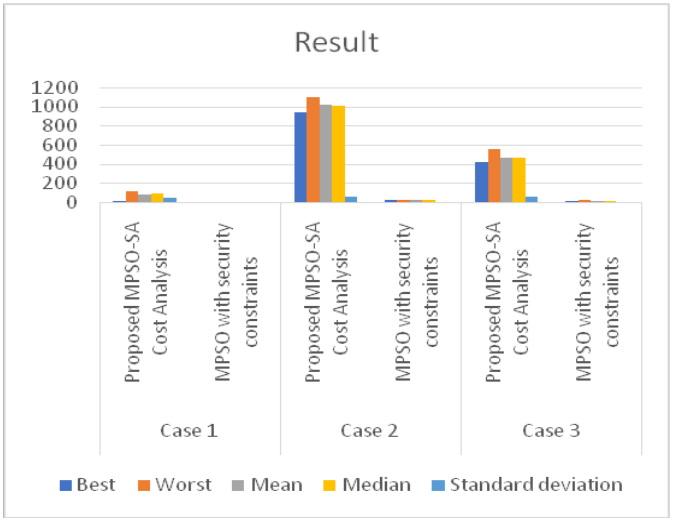

Figure 6

CONCLUSIONS

Diverse calculation dependent on the planning of assignment is thought about right now on various parameters like scheduling techniques, scheduling parameters, scheduling factors, discoveries, and situations, Right now, security fundamental model is point by point with occupation scheduling issue and its show is checked and separated and natural figuring, PSO and VNPSO calculation. The demonstration of the proposed MPSO-SA is inquired about with the three fundamental properties of organizing part, affiliation, cost statics, and risk probability statics. The starter consequences of the proposed MPSO-SA are separated and GA, PSO, and VNPSO and the outcome display that it gives a better cost limit and all-around safe likelihood and the mean estimation of each case are 40.74%, 12.45% and 34.01% superior to the current systems.

Future Scope

Scheduling is one of the most famous problems in cloud computing so; there is dependably a shot of adjustment of beforehand completed work in this specific field. The researchers at their own particular time played out their work according to as far as anyone is concerned space and after some time their work had been done some other people. During scheduling, they had considered various techniques and connected limitations. However, as the cloud computing is excessively huge that they had not possessed the capacity to catch all aspects in the meantime, yet they specified these certainties that there is a shot of adjustment of algorithms and which part needs to be modified.

REFERENCES

1. *T. Dillon, C. Wu, and E. Chang, "Cloud Computing: Issues and Challenges," Proceedings of the IEEE 24th International Conference Advanced Information Networking and Applications, Perth, 20-23 April 2010, pp.27–33.*
2. *F. Baroncelli, B. Martini and P. Castoldi, "Network Virtualization for Cloud Computing," Journal of Annals of Telecommunications, Vol. 65, No. 1-12, 2010, pp. 713–721.*
3. *T. D. Braun, H. J. Siegal, N. Beck, L. L. Boloni, M.Maheswaran, A. I. Reuther, J. P. Robertson, M. D. Theys, Y. Bin, D. Hensgen, and R. F.Freund, "A Comparison Study of Static Mapping Heuristics for a Class of Meta-Tasks on Heterogeneous Computing Systems," Proceedings of the 8th Heterogeneous Computing Workshop, San Juan, 12 April 1999, pp. 15–29.*
4. *M. Maheswaran, S. Ali, H. J. Siegal, D. Hensgen and R.F. Freund, "Dynamic Mapping of a Class of Independent Tasks onto Heterogeneous Computing Systems," Journal of Parallel and Distributed Computing: Special Issue on Software Support for Distributed Computing, Vol. 59, No. 2, 1999, pp. 107–131.*
5. *J. D. Ullman, "NP-Complete Scheduling Problems," Journal of Computer System Sciences, Vol. 10, No. 3, 1975, pp. 384–393.*
6. *JiahuiJin, Junzhou Luo, Aibo Song, Fang Dong, Runqun Xiong, "BAR: An Efficient Data Locality Driven Task Scheduling Algorithm for Cloud Computing", IEEE, 2011.*
7. *Xin Lu, ZilongGu, "A load-adaptive cloud resource scheduling model based on ant colony algorithm", IEEE, 2011*
8. *Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong, Wang, D., "Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization", IEEE, 2011.*
9. *Xiangqian Song, Lin Gao, Jieping Wang, "Job scheduling based on ant colony optimization in cloud computing", IEEE, 2011.*
10. *Zhang Bo, Gao Ji, Ai Jieqing, "Cloud Loading Balance algorithm", IEEE, 2011.*
11. *Van den Bossche, R., Vanmechelen, K., Broeckhove, J., "Cost-Efficient Scheduling Heuristics for Deadline Constrained Workloads on Hybrid Clouds", IEEE, 2012.*
12. *Frincu, M.E., Craciun, C., "Multi-objective Metaheuristics for Scheduling Applications with High Availability Requirements and Cost Constraints in Multi Cloud Environments", IEEE, 2012.*
13. *Jian Li, Sen Su, Xiang Cheng, Qingjia Huang, Zhongbao Zhang, "Cost-Conscious Scheduling for LargeGraph Processing in the Cloud", IEEE, 2011.*
14. *Chenhong Zhao, Shanshan Zhang, Qingfeng Liu, Jian Xie, Jicheng Hu, "Independent Tasks Scheduling Based on Genetic Algorithm in cloud computing", IEEE, 200.*
15. *Kai Zhu, Huaguang Song, Lijing Liu, Jinzhu Gao, Guojian Cheng, "Hybrid Genetic Algorithm for Cloud Computing Applications", IEEE, 2012.*
16. *Luohao Tang, Cheng Zhu, Weiming Zhang, Zhong Liu, "Robust COA planning with varying durations", IEEE, 2011.*

17. *Kessaci, Y., Melab, N., Talbi, E.-G., "A Pareto-based GA for scheduling HPC applications on distributed cloud infrastructures", IEEE, 2011.*

18. *Zhongni Zheng, Rui Wang, Hai Zhong, Xu.Zejie Zhang, "An approach for cloud resource scheduling based on Parallel Genetic Algorithm", IEEE, 2011.*

19. *GanGuo-Ning, Huang Ting-lei, GaoShuai, "Genetic simulated annealing algorithm for task scheduling based on cloud computing Environment", IEEE, 2010.*

20. *Diaz, C.O., Guzek, M., Pecero, J.E., Bouvry, P., Khan, S.U., "Scalable and Energy-Efficient Scheduling Techniques for Large-Scale Systems", IEEE, 2011.*

21. *Gao Ming and Hao Li, "An Improved Algorithm Based on Max-Min for Cloud Task Scheduling", Yunnam University, China, 2011.*

22. *Ching-Hsien Hsu, Tai-Lung Chen, "Adaptive Scheduling Based on Quality of Service in Heterogeneous Environments", IEEE, 2010.*

23. *Gutierrez-Garcia, J.O., Kwang Mong Sim, "A Family of Heuristics for Agent-Based Cloud Bag-of-Tasks Scheduling", IEEE, 2011.*

24. *Oprescu, A., Kielmann, T., "Bag-of-Tasks Scheduling under Budget Constraints", IEEE, 2011.*

25. *Netto, M.A.S., Buyya, R., "Offer-based scheduling of deadline-constrained Bag-of-Tasks applications for utility computing systems", IEEE 2009.*

26. *Hai Zhong1, 2, Kun Tao1, Xuejie Zhang1, 2," An Approach to Optimized Resource Scheduling Algorithm for Open-source Cloud Systems ", in Fifth Annual China Grid Conference, 2010.*

27. *Suraj Pandey, LinlinWu, Siddeswara Mayura Guru, Rajkumar Buyya," A Particle Swarm Optimization based Heuristic for Scheduling "Workflow Applications in Cloud Computing Environments".*

28. *Y. C. Lee, A. Y. Zomaya, Rescheduling for reliable job completion with the support of clouds, Future Generation Computer Systems 26 (2010)1192–1199.*

29. *V.M. Lo, "Task assignment in distributed systems", Ph.D. dissertation, Dep. Comput. Sci., Univ. Illinois, Oct. 1983.*

30. *G. Gharooni-fard, F. Moein-darbari, H. Deldari, and A. Morvaridi, Procedia Computer Science, Volume 1, Issue 1, May 2010, Pages1445-1454, ICCS 2010.*

31. *L. Zhang, Y.H. Chen, R.Y Sun, S. Jing, B. Yang. " A task scheduling algorithm based on PSO for Grid Computing", International Journal of Computational Intelligence Research. (2008), pp.37–43.*

A. Salman. "Particle swarm optimization for task assignment Problem". Microprocessors and Microsystems, November 2002. 26(8):363–371.

32. *Xu, M., Cui, L., Wang, H. and Bi, Y. ''A multiple QoS constrained scheduling strategy of multiple workflows for cloud computing'', IEEE11th Int'l Symposium on Parallel and Distributed Processing with Applications, Chengdu, China, pp. 629–634 (2009)*

33. *Topcuoglu, H., Hariri, S., & Wu, M. Y. (2002). Performance-effective and low-complexity task scheduling for heterogeneous computing. Parallel and Distributed Systems, IEEE Transactions on, 13(3), 260–274.*
34. *Salehi, M.A., and Buyya, R. ''Adapting market-oriented scheduling policies for cloud computing'', Proceedings of the 10th Int'l Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2010), Busan, Korea, pp. 351–362 (2010).*
35. *Yu, J., Buyya, R. and Tham, C.K. ''Cost-based scheduling of scientific workflow applications on utility grids'', First Int'l Conference on e-Science and Grid Computing, Melbourne, Australia, pp. 140–147(2005).*
36. *Sakellariou, R., Zhao, H., Tsiakkouri, E. and Dikaiakos, M.D. ''Scheduling workflows with budget constraints'', In Integrated Research in GRID Computing, S. Gorlatch and M. Danelutto, Eds Springer-Verlag., pp. 189–202, (2007).*
37. *Burke, E. K., McCollum, B., Meisels, A., Petrovic, S., & Qu, R. (2007). A graph-based hyper-heuristic for educational timetabling problems. European Journal of Operational Research, 176(1), 177–192.*
38. *Qu, R., Burke, E. K., McCollum, B., Merlot, L. T., & Lee, S. Y. (2009). A survey of search methodologies and automated system development for examination timetabling. Journal of scheduling, 12(1), 55–89.*
39. *Ochoa G, Qu R and Burke EK (2009). Analyzing the landscape of a graph based hyper heuristic for timetabling problems. In: Genetic and Evolutionary Computation Conference (GECCO 2009), ACM: New York, pp 341–348.*
40. *Tarek, Z., & Omara, F. A. (2014). Pso optimization algorithm for task scheduling on the cloud computing environment. Int. J. Comput. Technol, 13.*
41. *Li J, Burke EK and Qu R (2011). Integrating neural network and logistic regression to underpin hyper-heuristic search. Knowledge-based systems 24(2): 322–330.*
42. *Pillay N and Banzhaf W (2007). A genetic programming approach to the generation of hyper-heuristics for the uncapacitated examination timetabling problem. In: 13th Portuguese Conference on Artificial Intelligence (EPIA 2007), Lecture Notes in Computer Science, Vol. 4874, Springer: Berlin, pp 223–234.*
43. *Sabar NR, Ayob M, Qu R and Kendall G (2011). A graph coloring constructive hyper heuristic for examination timetabling problems. AppliedIntelligence 37(1): 1–11.*
44. *VivekManglani, Abhilasha Jain and Prof.VivekPrasad, Task Scheduling in Cloud Computing International Journal of Advanced Research in Computer Science,ISSN No. 0976–5697*
45. *P. Akilandeswari* and H. Srimathi, Survey and Analysis on Task Scheduling in Cloud Environment in Indian Journal of Science and Technology, Vol 9(37), ISSN (Print): 0974–6846 ISSN (Online): 0974-5645, October 2016.*
46. *P.Sanyasi Naidu and Babita Bhagat, Secure workflow scheduling in cloud environment using modified particle swarm optimization with scout adaptation, International Journal of Modeling, Simulation,and Scientific Computing, Vol. 9, No. 1 (2017) 1750064 (22 pages),World Scientific Publishing Company, 2017.*

47. *P.Sanyasi Naidu and Babita Bhagat, Modified Particle Swarm Optimization with Scout Adaptation and a New Inertia Weightfor Workflow Scheduling in Cloud Sector, Jour of Adv Research in Dynamical & Control Systems, Vol. 10, 10-Special Issue, 2018.*

48. *P.Sanyasi Naidu and Babita Bhagat,Workflow Scheduling under Secure Cloud Environment using MPSO-SA at International Journal of Recent Technology and Engineering (IJRTE) at ISSN: 2277-3878, Volume-8 Issue-3, September 2019.*